

Progettazione Automatica di Circuiti Elettronici

Docenti: Prof. Ing. Daniela De Venuto

Prof. Ing. Giovanni Mezzina

Relazioni di laboratorio A.A. 2020/2021

Studente: Francesco Caterina (Matricola 564974)

Esercitazione 7

Progettazione PCB RBBB-Pro

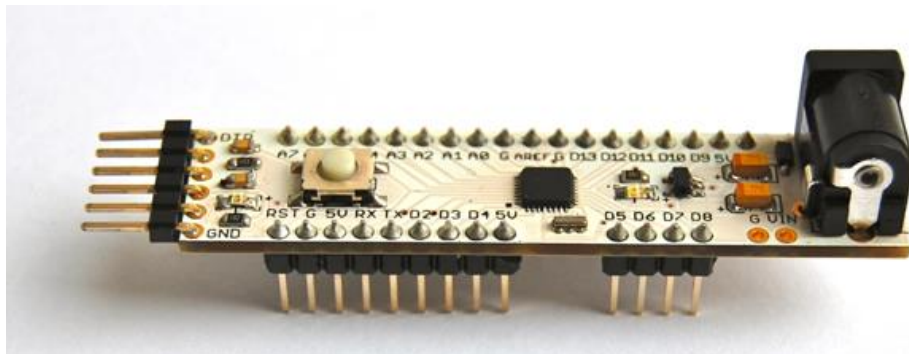
Contents

1	Introduzione ed obiettivo esperienza	2
2	Schema elettrico e breve analisi del progetto	3
2.1	Sezione di alimentazione	3
2.2	Sezione microcontroller	5
2.3	Sezione connettori	6
2.4	DRC, Smart PDF, BOM	7
3	Dallo schematico al PCB	9
3.1	Introduzione e setup iniziale di PCB Editor	9
3.2	Creazione area di lavoro, keep-in area, e component placement	10
3.3	Routing iniziale	12
3.4	Creazione dei "poligoni" per le linee di alimentazione +Vin, +5.0V	12
3.5	Copper pouring del piano di ground, thermal relief, clearance area	13
3.6	Completamento del routing di segnale e via holes	14
4	Ultimi ritocchi, generazione delle varie maschere, export file gerber	16
5	Conclusioni	20

1 Introduzione ed obiettivo esperienza

Obiettivo di quest' esperienza è imparare i concetti di base per la realizzazione di un PCB (Printed Circuit Board), i supporti sui quali al giorno d'oggi si realizzano tutti i circuiti elettronici a componenti discreti. Nello specifico utilizzeremo il software Orcad Capture per la realizzazione dello schematico, e Orcad PCB Editor (denominato Allegro nella versione con licenza) per la realizzazione del PCB.

Nello specifico andremo a realizzare la scheda RBBB Pro della Modern Devices; si tratta di una board clone di Arduino, che ha come core dunque il microcontrollore ATmega328P, ad 8 bit. Dunque è una board utile per lo sviluppo embedded di piccole applicazioni, per le quali non sia necessaria un'elevata potenza di calcolo. Sotto è riportato uno screen della scheda originale, sviluppata e realizzata dalla Modern Devices:



Risaltano subito sulla scheda alcuni componenti:

- Il power jack da 2.1mm x 5.5mm per l'alimentazione della scheda
- Il microcontrollore ATmega328P
- Il pulsante di reset, per riavviare l'esecuzione del codice (firmware) caricato sul micro
- I vari pin di I/O, tra i quali è incluso un connettore di interfaccia seriale utile per collegare la board al PC mediante un convertitore seriale/USB che utilizzi un chip FTDI

Cercheremo di replicare la scheda in tutte le sue funzionalità, mantenendo più o meno il layout originale. Obiettivo dell'esperienza resta comunque imparare i rudimenti necessari per la realizzazione di un PCB, e la teoria di base, dunque né il progetto in sé per sé, né lo specifico software utilizzato sono limitanti: tutti i concetti di base appresi in quest' esperienza valgono per la realizzazione di qualsiasi PCB, anche più complesso e di dimensioni maggiori.

Suddivideremo la realizzazione della scheda nelle seguenti fasi:

- Realizzazione dello schematico con Capture e breve analisi delle varie sezioni circuitali
- Realizzazione del PCB dallo schematico: component placement, creazione dei piani di ground e di alimentazione, routing
- Ultimi ritocchi, creazione maschere ed export gerber files

2 Schema elettrico e breve analisi del progetto

2.1 Sezione di alimentazione

La scheda è principalmente basata sul microcontrollore ATmega328P come già detto. Per la realizzazione dello schematico è stato utile in alcune fasi leggere il datasheet del microcontroller in questione. In particolare la prima cosa di cui ci dobbiamo occupare è l'alimentazione della scheda. Noi vogliamo che il microcontroller lavori a 16MHz; guardando il datasheet dell'ATmega328P nella sezione dell'alimentazione, notiamo il seguente grafico e tabella:

28.3 DC Characteristics

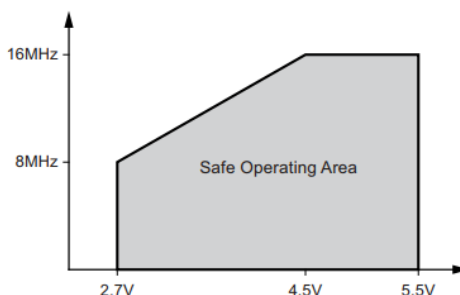
$T_A = -40^{\circ}\text{C}$ to $+125^{\circ}\text{C}$, $V_{CC} = 2.7\text{V}$ to 5.5V (unless otherwise noted)

Parameter	Condition	Symbol	Min.	Typ. ⁽²⁾	Max.	Units
Power supply current ⁽¹⁾	Active 4MHz, $V_{CC} = 3\text{V}$	I_{CC}		1.5	2.4	mA
	Active 8MHz, $V_{CC} = 5\text{V}$			5.2	10	mA
	Active 16MHz, $V_{CC} = 5\text{V}$			9.2	14	mA
	Idle 4MHz, $V_{CC} = 3\text{V}$			0.25	0.6	mA
	Idle 8MHz, $V_{CC} = 5\text{V}$			1.0	1.6	mA
	Idle 16MHz, $V_{CC} = 5\text{V}$			1.9	2.8	mA
Power-down mode ⁽³⁾	WDT enabled, $V_{CC} = 3\text{V}$	I_{CC}			44	μA
	WDT enabled, $V_{CC} = 5\text{V}$				66	μA
	WDT disabled, $V_{CC} = 3\text{V}$				40	μA
	WDT disabled, $V_{CC} = 5\text{V}$				60	μA

Notes: 1. Values with [Section 9.10 "Minimizing Power Consumption" on page 36](#) enabled (0xFF).
2. Typical values at 25°C .
3. The current consumption values include input leakage current.

28.4 Speed Grades

Figure 28-1. Maximum Frequency

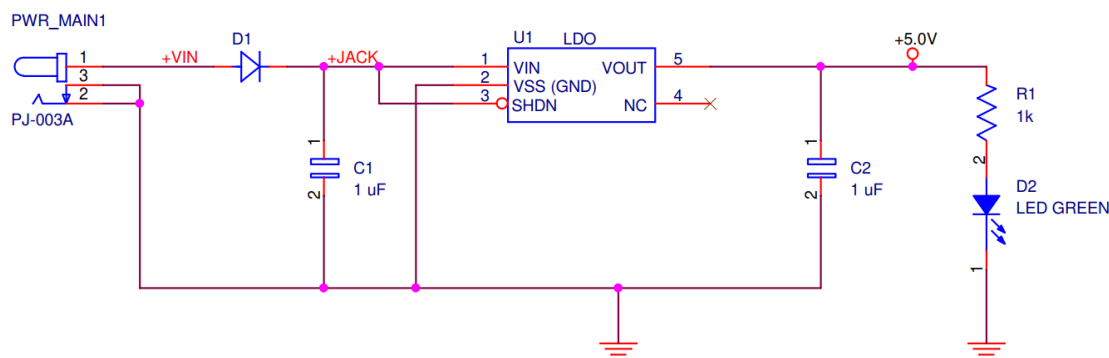


Come si può notare dal grafico, affinché il micro lavori a 16MHz e si resti comunque nella Safe Operating Area, lo si deve alimentare con una tensione dai 4.5V ai 5.5V. Si è scelta dunque una tensione di alimentazione standard da 5V, da dover generare dalla tensione di ingresso (che può essere superiore ai 5V, nel range 5V-10V per esempio) mediante un **regolatore di tensione**.

Notiamo inoltre, questa volta dalla tabella, che facendo operare il micro con un'alimentazione di 5V ed una frequenza di lavoro di 16MHz esso assorbe al massimo 14mA; tuttavia tale valore è riferito ad uno stato del microcontroller che lavora in modalità "low power", con molti dei suoi devices di I/O spenti (es ADC, UART, seriale, etc). Nella realtà il micro assorbirà una corrente di almeno 140mA, quando tutti i suoi devices sono abilitati.

Inoltre dobbiamo tener conto anche della corrente che verrà assorbita da eventuali periferiche connesse ai pin di I/O della scheda. Dunque per poter operare tranquillamente scegliamo un regolatore DC-DC che sia in grado di erogare almeno **300mA** di corrente massima, a 5V di alimentazione.

Si è optato per un regolatore lineare (LDO). Sotto riportiamo lo schema del circuito realizzato per la sezione di alimentazione, comprensivo di un diodo per impedire che la corrente scorra nel verso contrario (e bruci l'LDO) nel caso in cui si alimenti la scheda con polarità inversa, il chip scelto come regolatore lineare, la circuiteria necessaria al suo funzionamento ed un LED che indichi visivamente quando la scheda è connessa all'alimentazione:



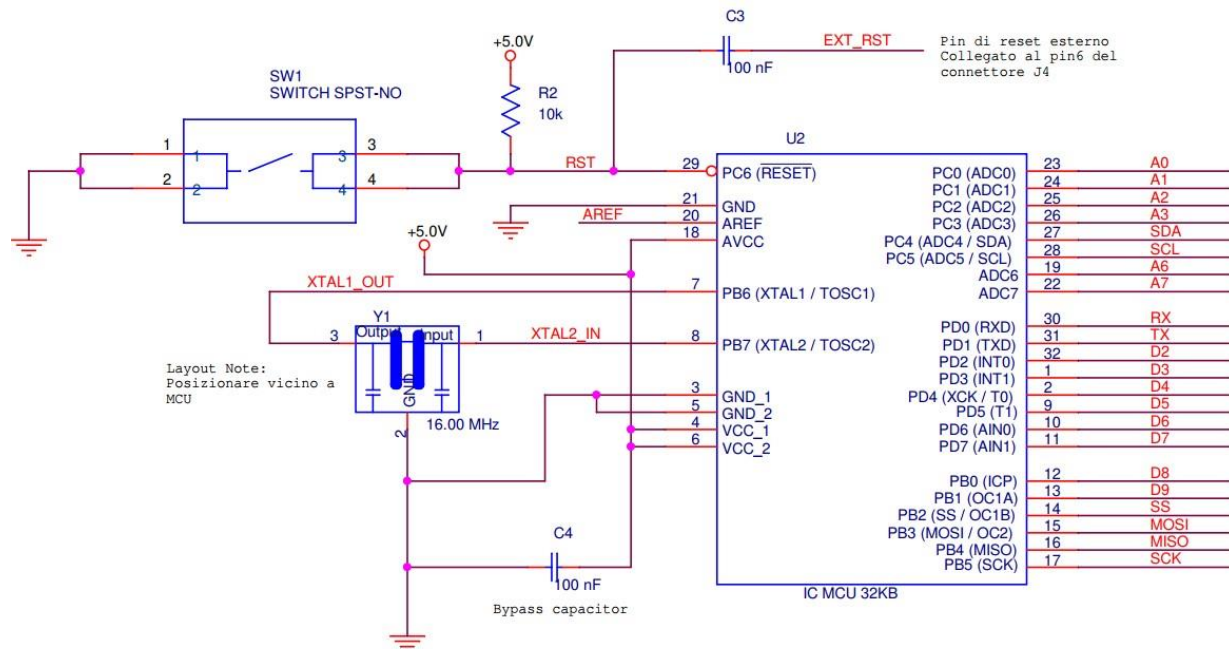
Il circuito è abbastanza semplice, in ingresso all'LDO c'è la tensione di alimentazione (che può variare in un range dai 6 ai 12V per esempio, dipende dalla tensione massima sopportabile in ingresso dall'LDO) e poi abbiamo le capacità C1, C2 sia in ingresso che in uscita per eliminare l'eventuale ripple e rumore presente sulla tensione di alimentazione e stabilizzare la tensione generata in uscita. Il nodo indicato dalla net 5.0V connesso all'uscita dell'LDO verrà utilizzato per alimentare l'intera scheda. Infine il pin 3 SHDN (Shutdown), che come si nota dal pallino in ingresso è negato, è stato connesso all'alimentazione positiva (valore logico 1) per mantenere l'LDO sempre acceso (avremmo per esempio potuto inserire uno switch a due vie, in questo caso, con rete di pull up, in modo da abilitare/disabilitare l'alimentazione semplicemente agendo sul valore logico del pin SHDN, spegnendo o accendendo l'LDO).

Nota: le capacità C1 e C2 non devono essere elettrolitiche, solitamente si scelgono capacità al tantalio (o ceramiche quando sono richiesti bassi valori di capacità) in quanto le capacità elettrolitiche hanno un valore elevato di ESR (Equivalent Series Resistance) e ciò non va bene quando vogliamo utilizzare i condensatori in ambito di elettronica di potenza per stabilizzare le tensioni di alimentazione, in quanto ad un ESR più elevato corrisponde un filtraggio peggiore della tensione di alimentazione (discorso simile per l'ESL, Equivalent Series Inductance).

Un altro trucco solitamente per risolvere questo problema è quello di mettere più capacità di basso valore, non elettrolitiche (es 100nF) in parallelo sull'alimentazione, magari su ogni IC, in modo che il valore di capacità in parallelo aumenti, e il valore di ESR complessivo diminuisca, garantendo un miglior filtraggio.

2.2 Sezione microcontroller

Seguendo la configurazione circuitale consigliata nel datasheet dell'ATMega328P abbiamo realizzato il seguente schematico per la sezione del microcontrollore:



Il micro è alimentato mediante l'alimentazione a +5.0V generata dall'LDO mediante i pin 4 e 6 (Vcc) ed i pin 3, 5 e 21 (GND). Si noti come tra Vcc e GND è stata posta un'ulteriore capacità C4 da 100nF per evitare che il micro, sensibile ad eventuali sbalzi di tensione, si resettasse involontariamente (se la tensione sul micro scende al di sotto di un certo valore infatti l'esecuzione del firmware potrebbe subire dei "glitch" o addirittura resettarsi completamente).

Il pin 29 invece è il pin di reset: quando esso viene portato a livello basso (0V) viene effettuato il reset del micro, e il codice da esso eseguito riparte dall'entry point (main). Tale pin di reset è stato collegato sia ad un sistema di reset manuale, realizzato mediante una resistenza di pull-up R2 da 10k ed uno switch tattile connesso verso ground, che ad una capacità C3 portata verso un pin esterno di External Reset.

Quando lo switch è aperto e non abbiamo alcun segnale sull'external reset la resistenza di pull-up R2 pone il pin 29 a Vcc, dunque livello logico 1, e non avviene alcun reset. Quando invece chiudiamo lo switch tattile verso ground il pin 29 viene portato a 0V, dunque livello logico 0, e avviene il reset del ciclo di esecuzione del micro (stessa cosa se arriva sull'external reset un segnale tale da portare il pin 29 a ground).

Sui pin 7 ed 8 invece è stato posto il quarzo, XTAL, comprensivo al suo interno già delle capacità necessarie a far oscillare l'oscillatore interno del microcontrollore. Il quarzo altro non è che un componente contenente appunto una lamina di quarzo, che come ben noto è un materiale **piezoelettrico**: quando viene applicata una differenza di potenziale tra le sue facce, esso subisce una deformazione elastica, mentre quando viene applicato uno stress meccanico allo stesso (pressione, deformazione, etc) tra le sue facce compare una differenza di potenziale. E' facilmente intuibile dunque come esso possa formare un sistema risonante, in quanto una variazione di una grandezza meccanica porta a una variazione di una grandezza

elettrica e viceversa: ciò può dar luogo, sotto le giuste condizioni, ad **oscillazioni che si autosostengono**. All'interno del micro è presente un oscillatore, che opportunatamente configurato può lavorare utilizzando un quarzo come elemento "reattivo". In base al tipo di quarzo scelto e alla configurazione interna dell'oscillatore del microcontrollore, cambierà la sua frequenza di funzionamento (quelle comuni per l'ATMega328P sono 4MHz, 8MHz, 16MHz). Si noti che il quarzo va messo il più vicino possibile al micro, in quanto le piste formano un induttanza (e capacità) parassita per unità di lunghezza, e se troppo lunghe esse vanno a spostare la frequenza di oscillazione del quarzo (che ricordiamo essere assimilabile ad un circuito RLC serie o parallelo; se aggiungiamo capacità o induttanze a tale circuito equivalente, la frequenza di oscillazione varia).

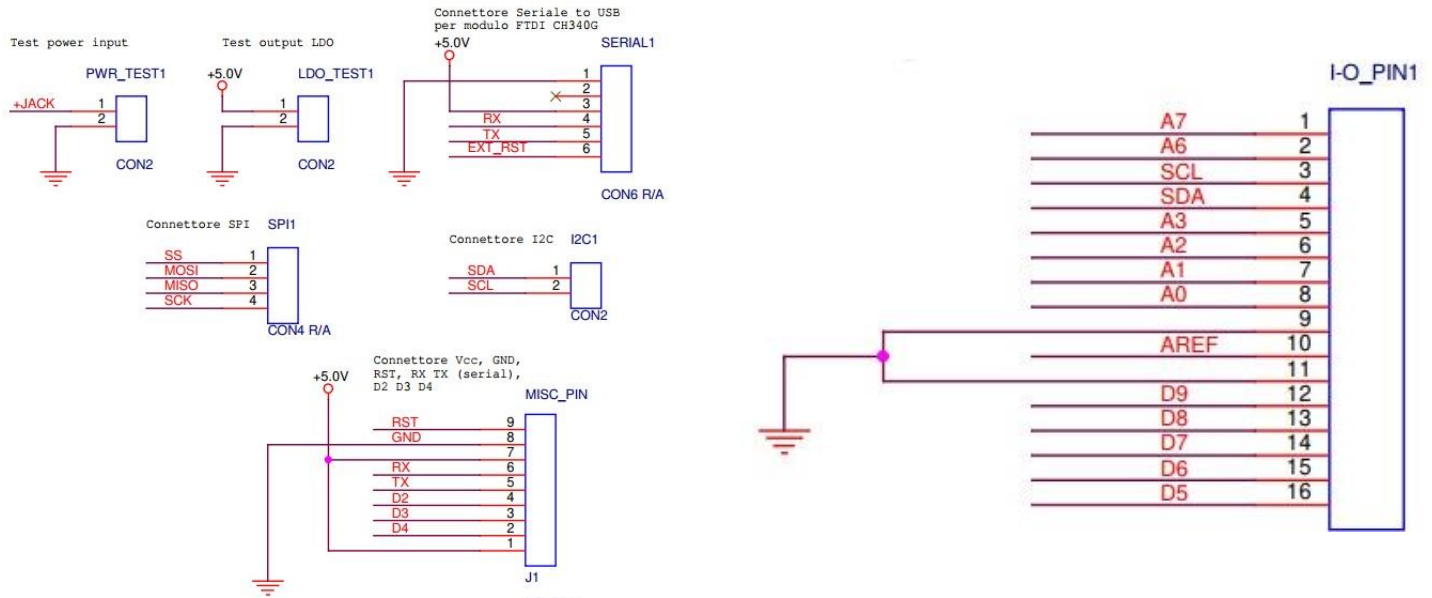
Infine, tutti i pin di I/O del microcontroller sono stati portati ai relativi connettori mediante net alias.

2.3 Sezione connettori

I connettori presenti sulla RBBB Pro originale, e che dunque vogliamo implementare nella nostra scheda, sono i seguenti:

- Connettore a due vie per il test dell'alimentazione di ingresso
- Connettore a due vie per il test della tensione generata dall'LDO (per testare eventuali malfunzionamenti dell'LDO, o per prelevare tensione qualora volessimo alimentare componenti che assorbono una piccola corrente)
- **Connettore "seriale" a 6 vie**, per portare alimentazione (Vcc, GND), segnale di reset esterno (EXT_RST) e bus seriali TX, RX ad un eventuale chip FTDI CH340G che permette la comunicazione con PC attraverso protocollo USB (i chip FTDI sono comunemente usati per la conversione da rs232, o qualsiasi protocollo seriale, a USB) in modo da poter caricare il firmware sul microcontroller e comunicare con esso via seriale, magari per il debug, attraverso porta COM virtuale.
- **Connettore SPI (Serial Peripheral Interface)** (a 4 vie) per connettere il micro ad altri IC digitali o sistemi esterni
- **Connettore I2C** (a 2 vie) utilizzato per connettere più dispositivi su uno stesso bus, in logica master/slave, utilizzato molto spesso nell'ambito dello sviluppo embedded per comunicare con dispositivi esterni quali display, sensori, etc
- Connettore a 9 vie per portare i primi I/O digitali del micro (D2-D4), il pin di reset RST e le linee RX TX (protocollo seriale) all'esterno
- Connettore a 16 vie, per portare all'esterno i restanti pin digitali (D5-D9), i pin analogici (A0-A7), e anche i pin dell'I2C (SDA, SCL) all'esterno, nonché il pin AREF utilizzato come riferimento di tensione per l'ADC interno del micro.

A pagina successiva riporto lo schematico realizzato per la sezione dei connettori, connessi mediante netalias ai relativi pin di I-O del micro e della scheda.



2.4 DRC, Smart PDF, BOM

Dopo aver completato tutte le sezioni dello schematico abbiamo effettuato il DRC su di esso (Design Rule Check) per controllare che non ci fossero errori di connessioni di natura elettrica (ovviamente il DRC non può rilevare errori relativi al design, ad esempio se usiamo un valore errato di resistenza o se connettiamo un transistor in maniera errata; rileva solo errori di natura elettrica, ad esempio eventuali cortocircuiti). Il DRC non ha fornito alcun errore, tranne i cortocircuiti realizzati (di proposito) tra i pin 3, 5 e 21 del micro (ground) e i pin 4, 6 e 18 (Vcc) che comunque erano voluti, in quanto tali pin del micro vanno connessi tutti alla relativa linea di alimentazione.

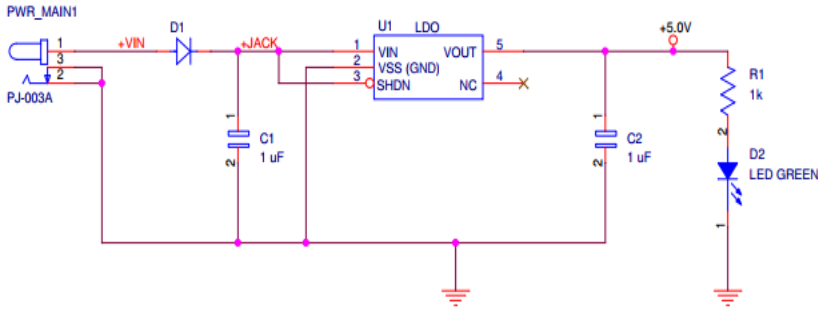
Inoltre abbiamo generato lo Smart PDF dell'intero schematico: esso permette di visualizzare innanzitutto l'intero schema che si è realizzato, con tutte le descrizioni e le caratteristiche dei vari componenti, ed inoltre permette di navigare nello schematico stesso mediante il menu che si utilizza solitamente per navigare nei PDF; ciò permette per esempio di individuare subito un determinato componente. Ciò risulta molto utile in schematici complessi di grande estensione. A pagina successiva riporto uno screen dello smart pdf generato.

Nota: per generare lo smart pdf è necessario installare GhostScript sul sistema.

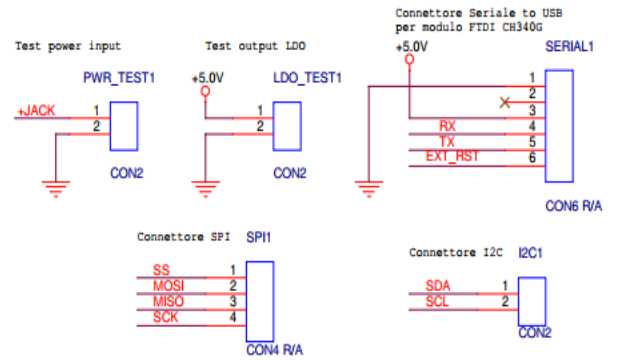
Assieme allo smart PDF a pagina successiva riporto la BOM (Bill Of Materials), generata sempre tramite Orcad Capture dallo schematico che abbiamo realizzato. Tale BOM è utilizzabile e importabile nei principali rivenditori di componenti elettronici (esempio DigiKey), in modo da visualizzare molto velocemente quali componenti sono disponibili subito, in magazzino, la quantità, etc. Se un determinato componente non è disponibile per esempio, DigiKey suggerisce altri componenti disponibile con i quali si può effettuare una sostituzione (da riportare poi ovviamente nel nostro schematico).

RBBB Pro - Modern Device - v1.0 - Poliba

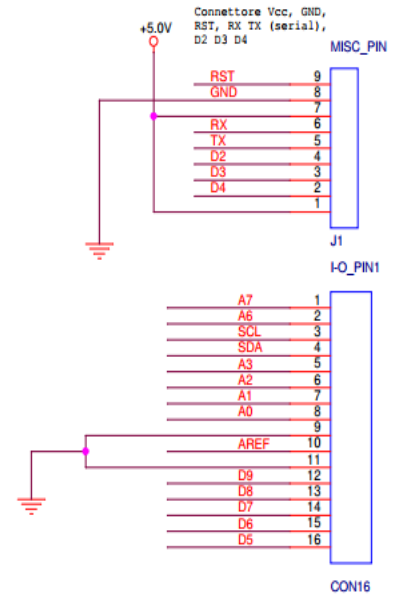
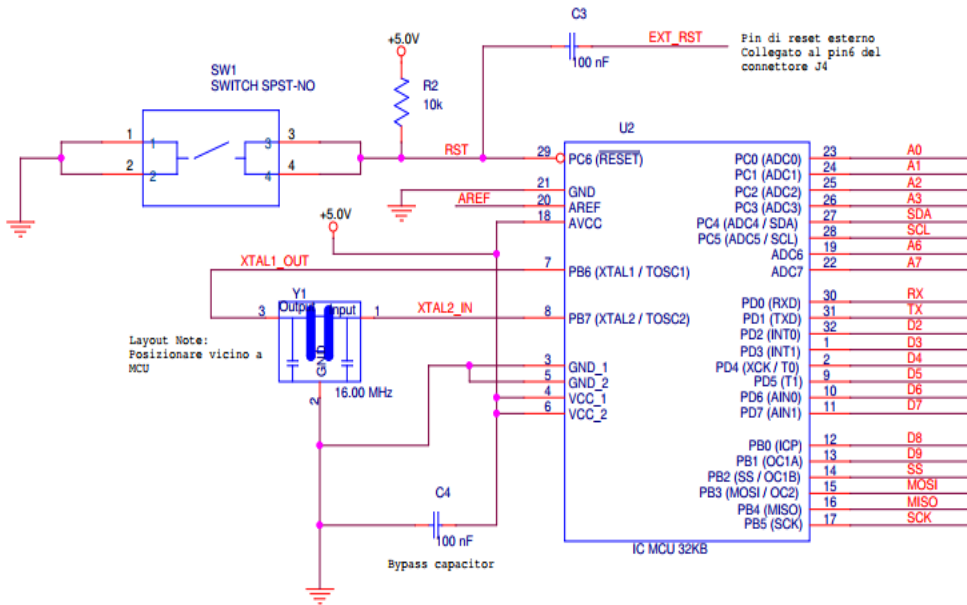
Power Management



Connectors



Microcontroller



PCB Design Course - Poliba		
Francesco Catarina Via E. Orsolina 4		
Title RBBB Pro - Modern Device		
Size A4	Document Number 0	Rev 1
Date: Thursday, July 14, 2022 Sheet 1 of 1		

Smart PDF dello schematico realizzato

Unique Part ID	Quantity	Part Reference	Value	Digi-Key Part Number
1	2	C1,C2	1 uF	1276-1066-1-ND
2	2	C3,C4	100 nF	1276-6557-1-ND
3	1	D1	DIODE	1N4148W-FDICT-ND
4	1	D2	LED GREEN	L62505CT-ND
5	1	I/O PIN	CON16	S1012EC-16-ND
6	3	I2C,PWR_TEST,LDO_TEST	CON2	S1012EC-02-ND
7	1	J7	MISC	S1012EC-09-ND
8	1	PWR_MAIN	PJ-003A	CP-003A-ND
9	1	R1	1k	RNCP0603FTD1K00CT-ND
10	1	R2	10k	RNCP0603FTD10K0CT-ND
11	1	SERIAL	CON6 R/A	732-5339-ND
12	1	SPI	CON4 R/A	S1012EC-04-ND
13	1	SW1	SWITCH SPST-NO	450-2055-2-ND
14	1	U1	LDO	MCP1802T-5002I/OTCT-ND
15	1	U2	IC MCU 32KB	ATMEGA328P-15AZTR-ND
16	1	Y1	16.00 MHz	535-9362-1-ND

Bill Of Materials (BOM)

3 Dallo schematico al PCB

3.1 Introduzione e setup iniziale di PCB Editor

Per realizzare il PCB a partire dallo schematico dobbiamo passare dal software Orcad Capture al software Orcad PCB Editor. Affinché essi siano “connessi” e sia possibile condividere informazioni tra loro (per esempio “trasportare” un componente dallo schematico in Capture alla zona di lavoro in PCB Editor), dobbiamo assicurarci che in Capture sia abilitata l’opzione “Enable Intertool Communication”.

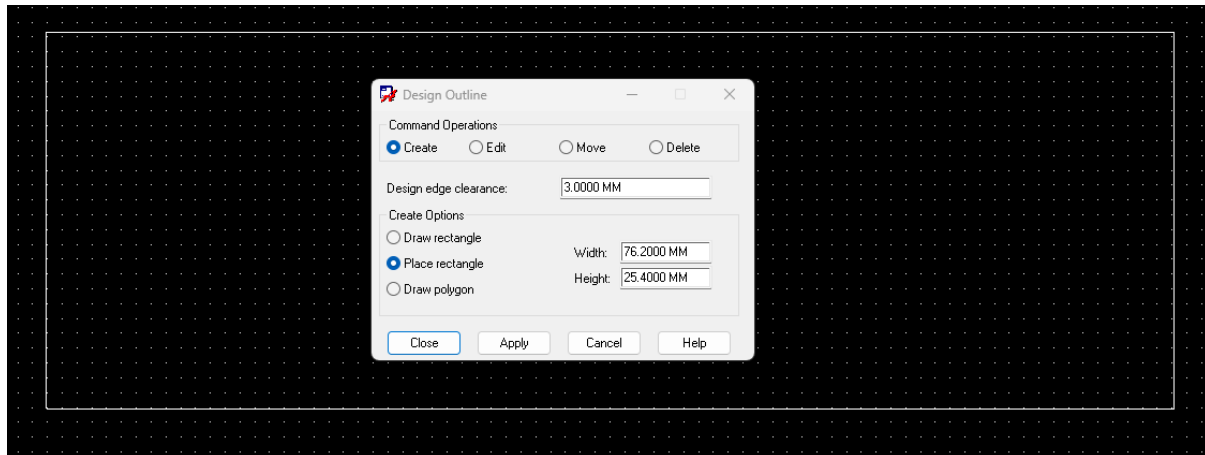
Inoltre in PCB Editor dobbiamo fare un breve setup iniziale, dato che lo utilizziamo per la prima volta, in cui impostiamo, tra le più importanti, le seguenti feature:

- L’unità di misura utilizzata (nel nostro caso abbiamo impostato millimetri; solitamente di default è in mils, cioè millesimo di pollice; l’equivalenza tra le due unità di misura è $1 \text{ mils} = 0.0254 \text{ mm}$)
- Le regole di copper pouring per l’eliminazione di eventuali isole di “dead copper”, isole di rame “morto” che non ha elettricamente alcuna utilità ai fini della connessione elettrica. Esse vengono a formarsi solitamente sui piani più estesi, cioè quelli dedicati al ground ed alle linee di alimentazione, quando vengono attraversate da altre piste
- La dimensione delle piste e la distanza minima tra di esse secondo la regolamentazione IPC2221. Queste grandezze vengono impostate nel Constraint Manager, e devono seguire oltre che la regolamentazione IPC2221 anche le regole per la manifatturabilità della scheda: se il produttore che andrà a produrre la scheda per esempio non ci può garantire una distanza di un certo valore tra le piste, dobbiamo sicuramente aumentare tale valore nel constraint manager, altrimenti qualora dovessimo realizzare piste con tale valore di distanza minima quando andremo a produrre la scheda risulteranno errori di natura elettrica (esempio cortocircuito tra due piste, oppure le due piste vengono fuse in un’unica pista) dovuta al fatto che non abbiamo rispettato i constraint dati da chi ci andrà a fabbricare la scheda. Discorso simile per lo spessore delle piste.
- La palette dei colori per individuare meglio alcuni componenti, pin, e le piste relative ad alcune net più importanti (+Vin, +5.0V, GND), nonché i colori dedicati alla soldermask, alla pastemask e al silkscreen (generate successivamente, quando abbiamo finito di realizzare l’intero PCB)
- Inoltre abbiamo abilitato l’opzione “NoRat” per la ratline di ground, in quanto la net GND avrà un suo piano interamente dedicato sul bottom, quindi non abbiamo bisogno delle ratlines per generare la pista di ground (anzi, sarebbero di intralcio).

Dopo aver effettuato questo setup iniziale, siamo pronti alla realizzazione vera e propria del progetto della scheda.

3.2 Creazione area di lavoro, keep-in area, e component placement

Iniziamo a definire l'area di lavoro, dal menu "Outline -> Place Rectangle" inserendo le dimensioni (altezza e larghezza) della scheda che intendiamo realizzare. Queste dimensioni poi potranno esser cambiate, se ci rendiamo conto che necessitiamo di più spazio (a meno di vincoli dati dal committente del progetto).

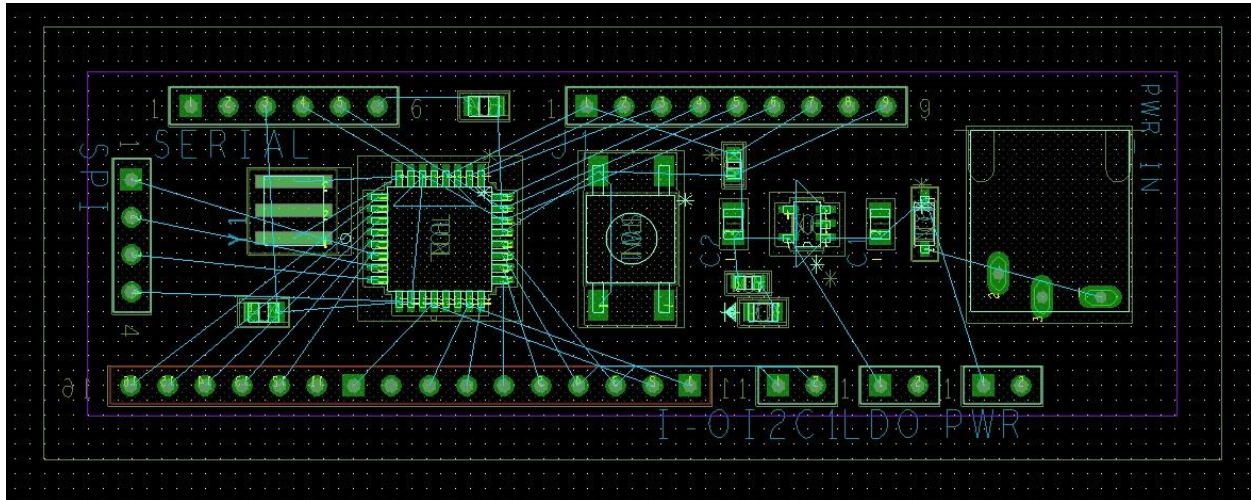


Dopo aver impostato la grandezza della scheda, definiamo il parametro "Design Edge Clearance" che permette di definire la keep-in area. La keep-in area è un'area entro la quale dobbiamo mantenere il piazzamento di tutti i nostri componenti, sia per motivi di **compatibilità elettromagnetica** (evitare interferenze con altri PCB montati vicino a quello che stiamo progettando, per esempio), sia per motivi di natura **elettrostatica**: qualora il PCB sia esposto all'ambiente esterno, e venga ad esempio a contatto con oggetti o persone cariche elettrostaticamente, potrebbe verificarsi una scarica elettrostatica su alcuni componenti, e potrebbero danneggiarsi (**i circuiti integrati CMOS sono particolarmente sensibili alle scariche elettrostatiche**). La keep-in area, mantenendo tutti i componenti al suo interno, permette di "isolare" i componenti dall'ambiente esterno, fornendo un certo margine dal bordo del PCB alla zona in cui piazziamo i componenti, ed in tal modo mitigheremo questi fenomeni.

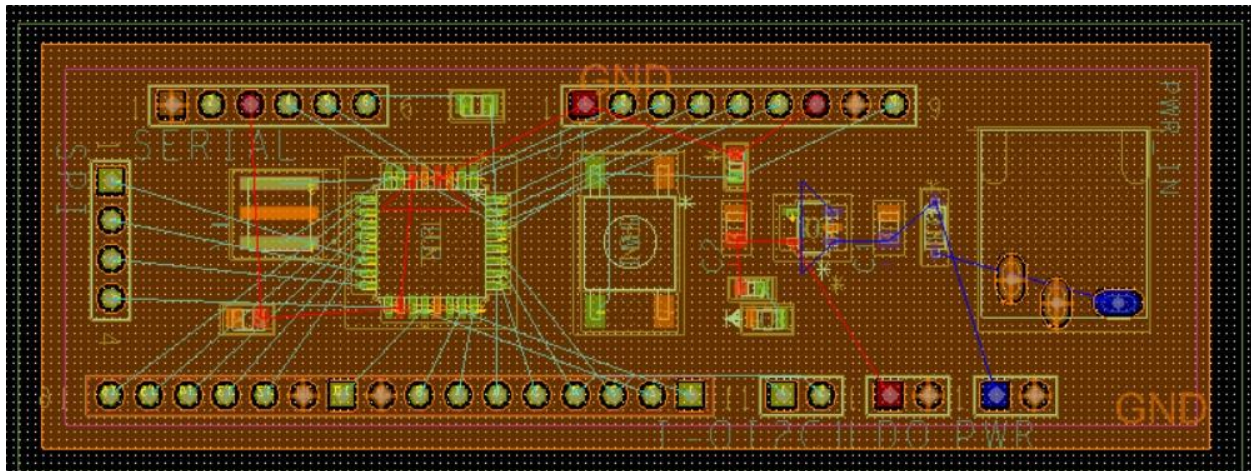


Area PCB (verde), e keep-in area (arancione)

Dopo tale setup, possiamo iniziare a piazzare tutti i componenti nella keep-in area, avendo ovviamente l'accuratezza di mettere vicino i componenti che fanno parte della stessa sezione. Praticamente nel PCB andremo a ricreare le varie sezioni che abbiamo creato nello schematico, interconnettendole poi tra loro. Si ricordi che il placement di alcuni componenti è critico, come ad esempio il risonatore a quarzo, che dev'essere posto vicino al microcontrollore. Il risultato finale, dopo aver completato il placement, è il seguente:



Sotto, lo stesso placement, dopo aver assegnato alle varie net più importanti determinati colori per identificarle meglio:



Placement con net identificate mediante colori. Arancione per il ground, rosso per +5.0V, blu per +Vin

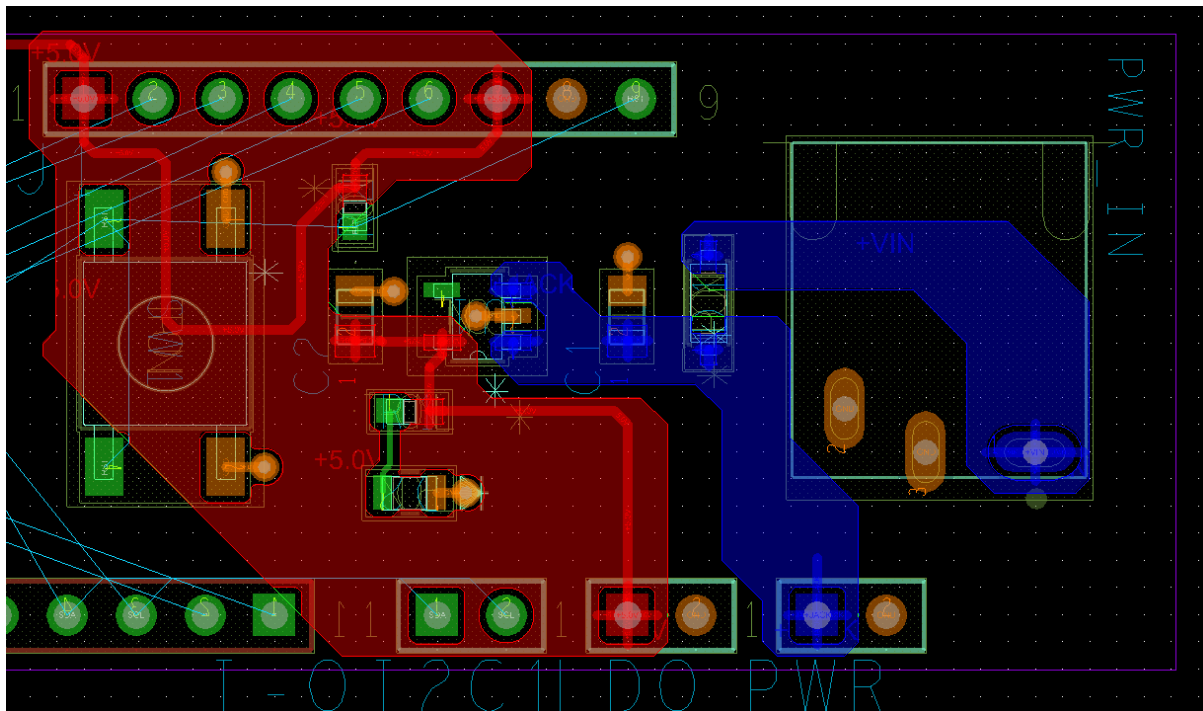
Nella figura superiore è già stato creato il piano di ground sul lato bottom della scheda (motivo per cui è di colore arancione). La scheda che stiamo realizzando infatti è a due layer, un top ed un bottom. Sul layer top andremo a posizionare tutti i componenti (sia SMD che Through Hole), ed inoltre andremo ad effettuare la maggior parte del routing, mentre sul layer bottom avremo il piano di ground, le saldature dei componenti through hole, i via hole (che passano dal layer top al bottom) e il resto del routing che non riusciamo a completare sul lato top: le piste di routing realizzate sul bottom andranno ad interrompere il piano di ground, per tale motivo è consigliabile realizzarne in numero minimo.

3.3 Routing iniziale

Come si può notare tra i vari pin dei vari componenti che abbiamo piazzato sono comparse delle linee: esse sono denominate **ratlines** e ci dicono come dobbiamo collegare elettricamente, con le piste, i vari pin per far in modo che il PCB realizzato sia elettricamente equivalente al nostro schematico. Tali informazioni sono disponibili al software PCB Editor in quanto quando abbiamo importato il progetto da Capture, esso ha condiviso con PCB Editor anche la netlist dello schematico. Nella netlist sono presenti tutte le informazioni su tutte le connessioni effettuate nello schematico (anche quelle realizzate tramite netalias). Unendo le varie ratlines con le piste iniziamo ad effettuare un primo routing iniziale. Si noti che il routing va effettuato in modalità **"etch edit"**, (etching = incisione, asportazione) così denominata in Orcad PCB Editor in quanto creando le piste stiamo effettivamente asportando del rame dalla basetta per la creazione delle piste (che sia mediante fotoincisione o mediante macchina CNC).

3.4 Creazione dei "poligoni" per le linee di alimentazione +Vin, +5.0V

Le "piste" che sulle quali passerà maggior corrente, e dunque che necessitano di esser più larghe sia per fornire meno resistenza al percorso di corrente elettrica, sia per dissipare più potenza, sono quelle dedicate alle linee di alimentazione: GND, +Vin e +5.0V. Mentre il ground ha tutto un piano a sé dedicato, le linee +Vin e +5.0V sono state realizzate per il momento con delle semplici piste. Per questo dobbiamo "rinforzarle" allargandole, creando dei poligoni attorno alle linee originali mediante la funzione "shapes" e "shape edit" di PCB Editor. Sotto è riportato un esempio di come diventano le linee di alimentazione una volta allargate utilizzando i poligoni:



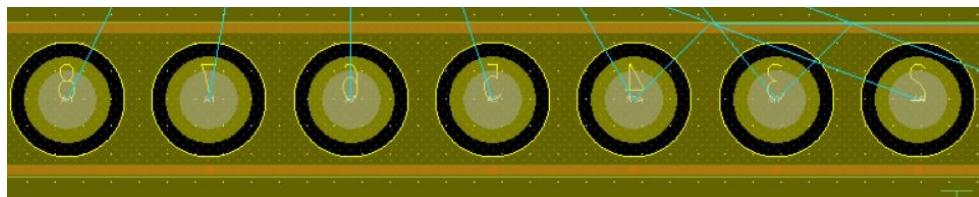
Piano +5.0V rosso, piano +Vin blu

Adesso le piste di alimentazione essendo più larghe potranno trasportare più corrente, e allo stesso tempo riusciranno a dissipare meglio il calore (avendo un'estensione maggiore).

3.5 Copper pouring del piano di ground, thermal relief, clearance area

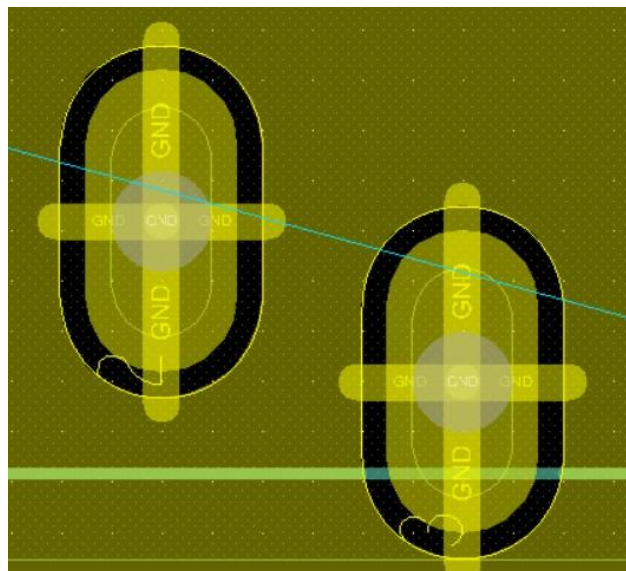
Oltre a ciò dobbiamo creare il piano di ground sul lato bottom, mediante la tecnica di **copper pouring**. Consecutivamente alla creazione del piano di ground sul lato bottom, il software creerà automaticamente i thermal relief e le clearance area.

La clearance area è una **zona di "vuoto" creata per esempio attorno ai pin through hole che non devono essere elettricamente in contatto col piano di ground** (o comunque con altri piani conduttori su cui si vengono a trovare tali pin, per esempio il piano +5.0V sul lato top). Viene praticamente asportata la totalità del rame attorno al pin, in modo da garantirne l'isolamento elettrico. Sotto un esempio di varie clearance area, creatasi sul piano bottom attorno ai pin di uno dei connettori dopo il copper pouring del piano di ground. In tal modo tali pin saranno elettricamente isolati dal piano su cui sono posti, e sarà possibile effettuarne il routing mediante piste verso i pin del microcontroller senza cortocircuiti verso i piani di ground o alimentazione, essendovi isolamento elettrico tra i pin in questione e i piani conduttori.



Clearance area

Il thermal relief (o freno termico) è un espediente utilizzato sui pin Through Hole saldati sulla scheda. Esso è necessario affinché in fase di saldatura il calore non si disperda verso tutta la pista di rame, ma resti concentrato sul pin che intendiamo saldare, rendendo così più facile la saldatura del pin del componente in questione. Sotto un esempio di thermal relief generato quando andiamo a creare il piano di ground mediante il copper pouring. Ovviamente possiamo avere thermal relief anche su qualsiasi piano associato all'alimentazione, per esempio su un pin through hole connesso al piano +5.0V.



Thermal Relief

Il thermal relief praticamente consiste in una clearance area, che isola termicamente il pin dal piano conduttore, a cui viene sovrapposta poi una croce di rame, che permette la conduzione elettrica minimizzando quella termica.

In tal modo la connessione elettrica con il piano di ground è sempre garantita, ma si limita notevolmente la diffusione del calore dal pin verso il resto del piano di ground, migliorandone l'isolamento termico in quanto abbiamo asportato del rame.

3.6 Completamento del routing di segnale e via holes

Abbiamo completato il routing di tutte le piste (piani) di alimentazione, ground, +5.0V, +Vin. Resta ora da fare il routing di "segnale".

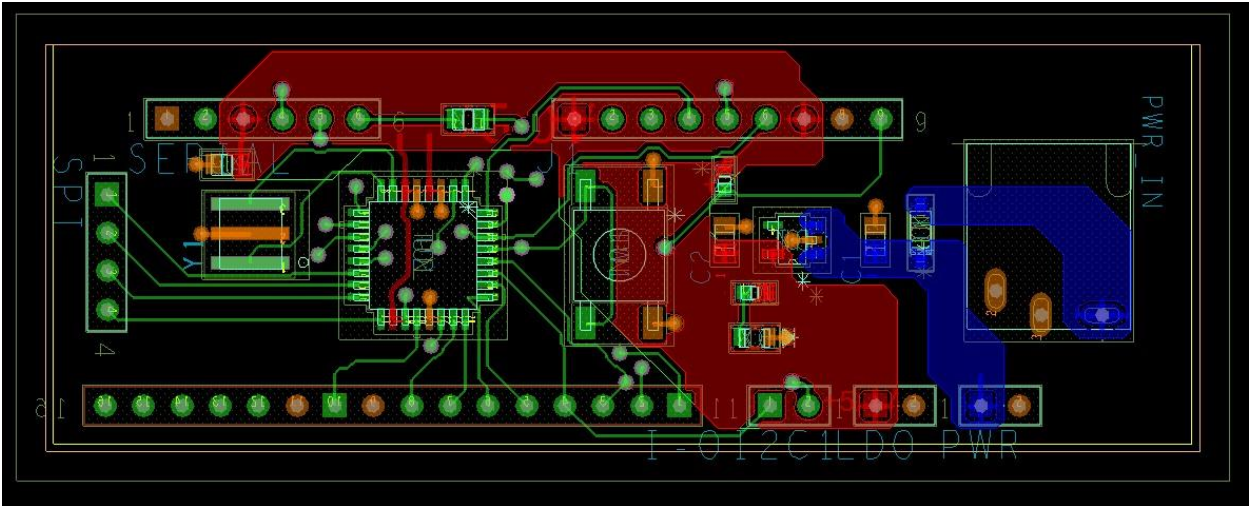
A tal fine iniziamo a collegare dapprima alcune connessioni sul top layer (tutto ciò sempre in modalità etch edit, in quanto stiamo intagliando il layer di rame). Si noti che potremo realizzare piste di segnale anche attraversando i piani di alimentazione, in quanto il software automaticamente isola le piste che stiamo creando dal piano di alimentazione che stiamo attraversando, creando una specie di clearance area tra le piste di segnale appena create e il piano di alimentazione che stiamo attraversando, isolandole dunque elettricamente.

Tuttavia dopo un po' ci accorgeremo che non potremo più effettuare connessioni sul top layer, in quanto ulteriori piste andrebbero ad incrociarsi con quelle già create, che non possiamo interrompere. La soluzione è quella di portare in parte le piste in questione sul lato bottom, dove c'è il piano di ground (anche questo verrà isolato comunque dalle piste che stiamo creando), attraverso i **via holes**. Questi ultimi altri non sono che connessioni elettriche tra il top layer e il bottom layer, realizzate per esempio mediante un foro sulla pcb, metallizzato al suo interno e su un piccolo intorno del foro su entrambi i layer, in modo da **garantire conduttività elettrica tra top layer e bottom layer nel punto in cui abbiamo piazzato il via hole**.

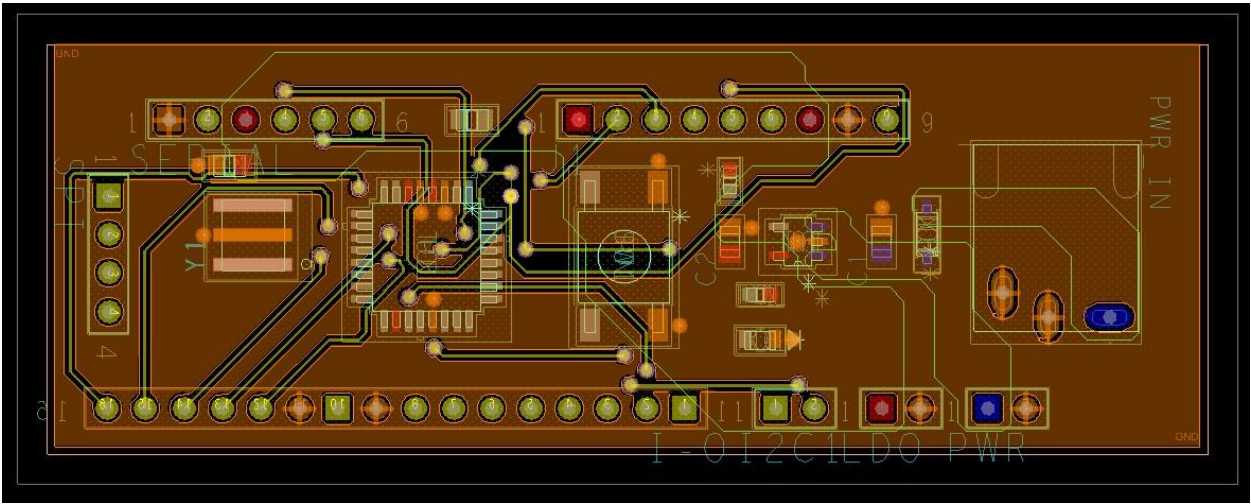
In tal modo quando non riusciamo a completare il routing da un pin ad un altro sul top layer perché ci sono altre piste di mezzo che creerebbero incroci (assolutamente da evitare, ovviamente), possiamo portare tale pista sul bottom layer attraverso un via hole, continuando lì il routing, e successivamente dopo aver "evitato" tutte le piste presenti sul top layer che ci davano fastidio possiamo risalire sul top layer mediante un ulteriore via hole, e portare al termine la nostra connessione elettrica verso il pin desiderato. Ovviamente qualora il pin da raggiungere sia quello di un componente through hole possiamo restare sul bottom layer e raggiungere direttamente il pin desiderato; se invece si tratta di un pin di un componente SMD invece, posizionato dunque sul top layer, dovremo necessariamente risalire.

E' consigliabile comunque mantenere la lunghezza delle piste sul bottom layer abbastanza corta, ed esiguo in numero rispetto a quelle del top layer, in modo da non interrompere troppo spesso il piano di ground. Questo è caldamente consigliabile soprattutto per circuiti high-speed che lavorano ad alta frequenza, dove il piano di ground agisce effettivamente da "schermatura" per tutti gli altri segnali, riducendo al minimo problemi di compatibilità elettromagnetica, quali crosstalk e interferenze. Nel nostro circuito tutto ciò non rappresenta un grosso problema, in quanto lavora a frequenze relativamente basse, dunque possiamo mantenere una lunghezza media delle piste sul lato bottom (senza esagerare ovviamente).

Dopo aver completato tutto il routing il nostro PCB si presenta nella seguente maniera; ho suddiviso la rappresentazione del top layer da quella del bottom layer, in maniera da rendere il tutto più chiaro.



Top Layer



Bottom Layer

Si notino i piani di alimentazione +Vin (in blu) e +5.0V (in rosso) sul top layer, nonché il routing di segnale (le restanti piste in verde); sul bottom layer invece notiamo il piano di ground (in arancione) interrotto dalle varie piste di routing di segnale che completano le restanti connessioni elettriche. Effettivamente sarebbe stato possibile ridurre il numero delle piste sul bottom layer e la loro lunghezza, per ridurre al minimo le interruzioni del piano di ground, scegliendo un diverso placement dei componenti magari, o una rinumerazione dei pin dei connettori, ma trattandosi di un progetto a livello didattico il risultato ottenuto è comunque soddisfacente e comunque le connessioni elettriche sono garantite senza errori di qual sorta, come può mostrare il risultato ottenuto dal report DRC (Design Rule Check) effettuato questa volta in PCB Editor:

Design Name C:/Users/FraH/Desktop/VirtualDesktop/00-University/ProgettazAutom/00a-Relazioni/07-PCB/RB8B_PCB_Project/allegro/RB8B_PCB.brd
Date Sat Jul 16 20:24:57 2022

DRC Error Count Summary	
DRC Error Type	DRC Error Count
Total DRC Errors	0

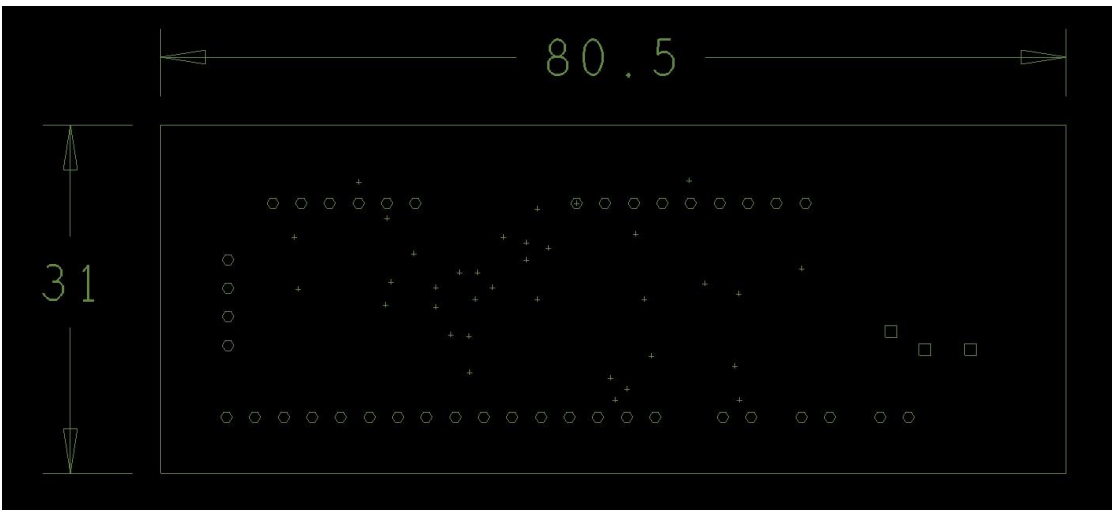
Detailed DRC Errors						
Constraint Name	DRC Marker Location	Required Value	Actual Value	Constraint Source	Constraint Source Type	Element 1 Element 2

Total Unconnected Pin Pairs: 0

In esso dovrà essere integrata anche la **drill table** che permette al produttore di identificare i vari tipi di fori da realizzare sulla scheda, nonché la maschera per il layout dei fori che dice al produttore proprio in quale punto della scheda andare ad effettuare un certo determinato tipo di fori:

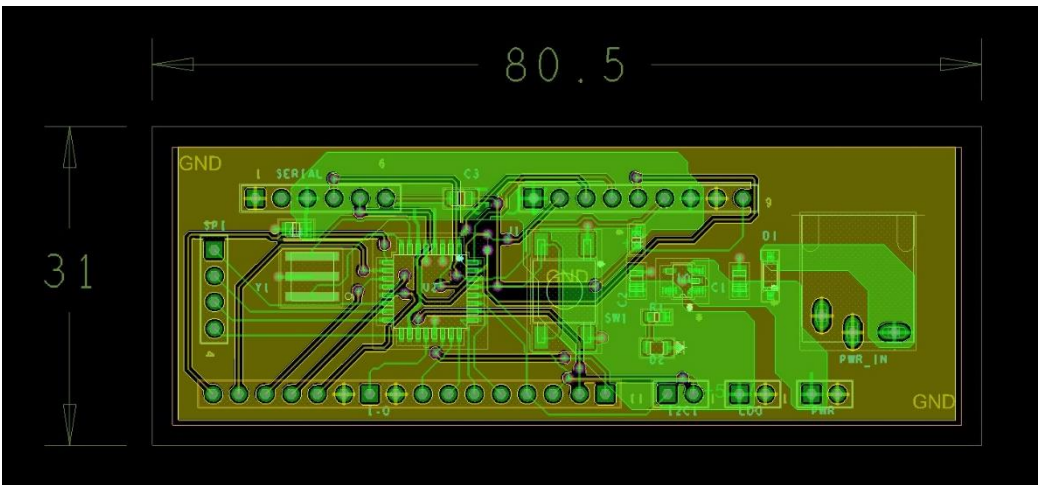
DRILL CHART: TOP to BOTTOM			
ALL UNITS ARE IN MILLIMETERS			
FIGURE	FINISHED_SIZE	PLATED	QTY
.	0.5334	PLATED	36
□	1.0	PLATED	3
○	1.0668	PLATED	41

Drill Table



Drill mask layout

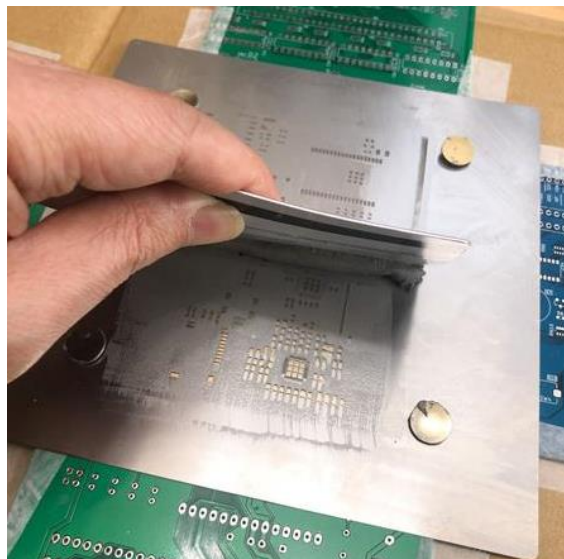
Si noti come ogni tipo di foro presente sulla drill mask ha una particolare forma, le cui caratteristiche sono riportate nella leggenda della drill table. In tal modo il produttore saprà che tipo di perforazione realizzare in tale punto del PCB. Dopo aver aggiunto le quote delle dimensioni della scheda, sovrapposto la visualizzazione top layer/bottom layer e eliminato i vari colori che distinguevano le varie net, il nostro progetto si presenta così:



Le dimensioni sono state leggermente aumentate rispetto al canvas iniziale che si era creato. Si noti come sia stata mantenuta comunque la keep-in area di 3mm dal bordo della scheda. Si noti inoltre come le piste dedicate al top layer siano in verde, mentre quelle dedicate al bottom layer siano in giallo.

Dopodiché ci restano da generare:

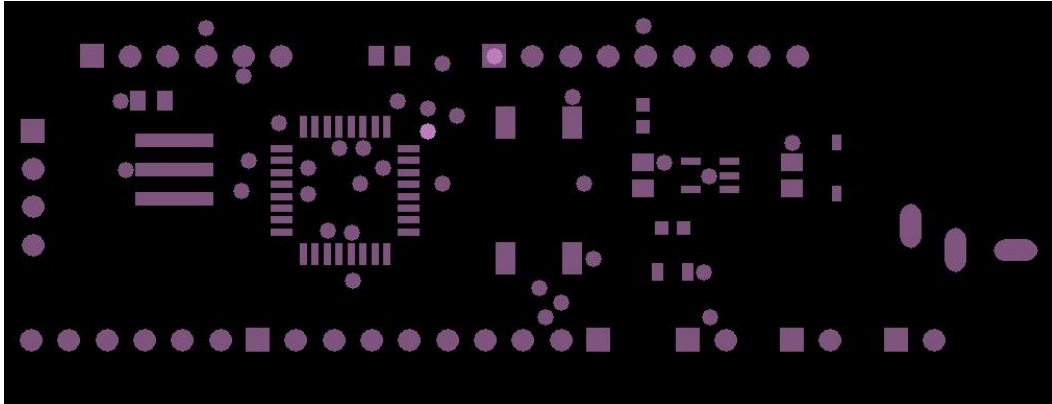
- Il layout complementare alla soldermask top e alla soldermask bottom. La soldermask in generale è uno strato di resina o polimero isolante (solitamente di color verde, che dà il caratteristico colore alle schede elettroniche, ma può esser anche rosso o di altri colori) che viene posta su tutta la scheda per isolare tutte le piste di rame. Tuttavia se andassimo a ricoprire anche i pad per saldare i componenti SMD e le piazzole dei componenti through hole con la soldermask, essi non sarebbero più conduttivi e non avremmo conduzione elettrica tra il pin del componente e il pcb, oltre ad essere praticamente impossibile effettuare la saldatura sul polimero della soldermask in quanto non è un metallo. Per ovviare a ciò appunto si generano i layout complementari alla soldermask, che vanno praticamente ad evitare che i pad e le piazzole su cui dobbiamo saldare dei componenti vengano “risparmiati” dalla soldermask; in tal modo il rame in tali punti resterà “nudo” e sarà possibile saldare i componenti in tali punti, mantenendo isolate elettricamente dall’ambiente esterno le piste. OSS: La soldermask funge sia da isolante elettrico per le piste, sia da strato protettivo, per evitare che il rame si ossidi e le performance conduttive vengano degradate.
- La pastemask, denominata anche **stencil**: essa è una maschera metallica che verrà generata, in cui sono coperte tutte le aree della scheda tranne i punti in cui sono presenti i pad e le piazzole su cui dobbiamo saldare. In tal modo potremo ricoprire tali punti con la pasta saldante, evitando che essa vada sulle piste o su altre zone indesiderata. Sotto, in foto, un esempio di pastemask utilizzata durante il processo di passaggio della pasta saldante sul PCB per preparare la scheda alla saldatura dei componenti.



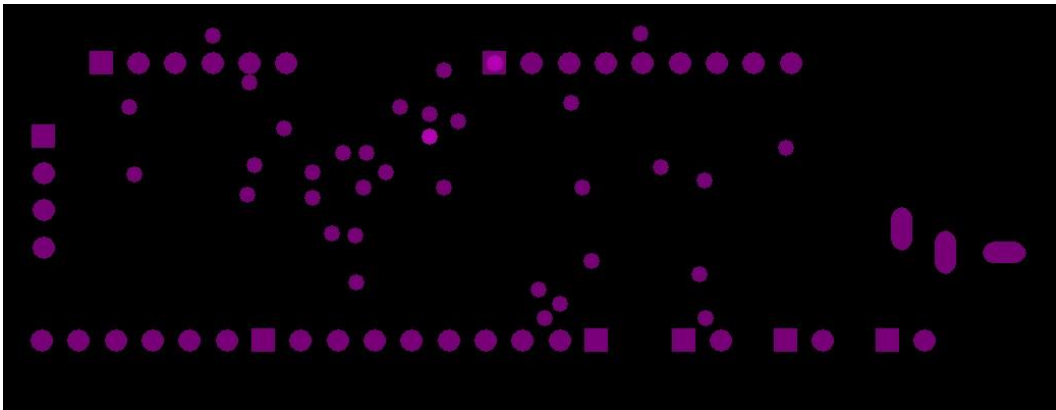
Si noti che i fori della pastemask si troveranno solo in corrispondenza dei componenti SMD, in quanto quelli through hole vengono saldati sul bottom layer utilizzando un normale saldatore, e non un saldatore ad aria calda come nel caso dei componenti SMD.

- Restano da generare infine l'**assembly top**, che ci dirà come vanno posizionati i componenti (in questo caso solo sul lato top, in quanto abbiamo componenti solo sul top layer) e il **silkscreen**, che dà denominazione ai vari componenti e verrà stampato sulla scheda tramite un inchiostro bianco, in modo da sapere quale componente saldare in ciascuna posizione (es R1, R2, U1, U2, etc).

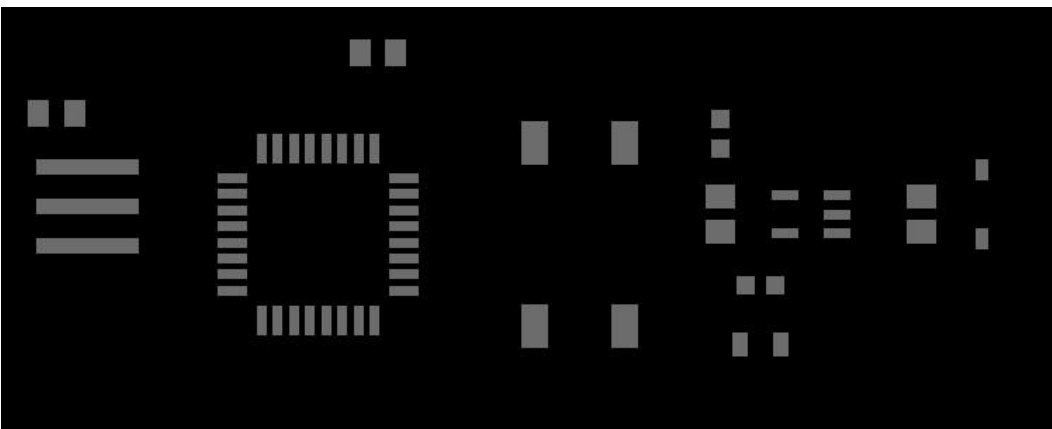
Sotto riporto la preview delle varie maschere generate per il progetto.



Top layer (complementary) soldermask

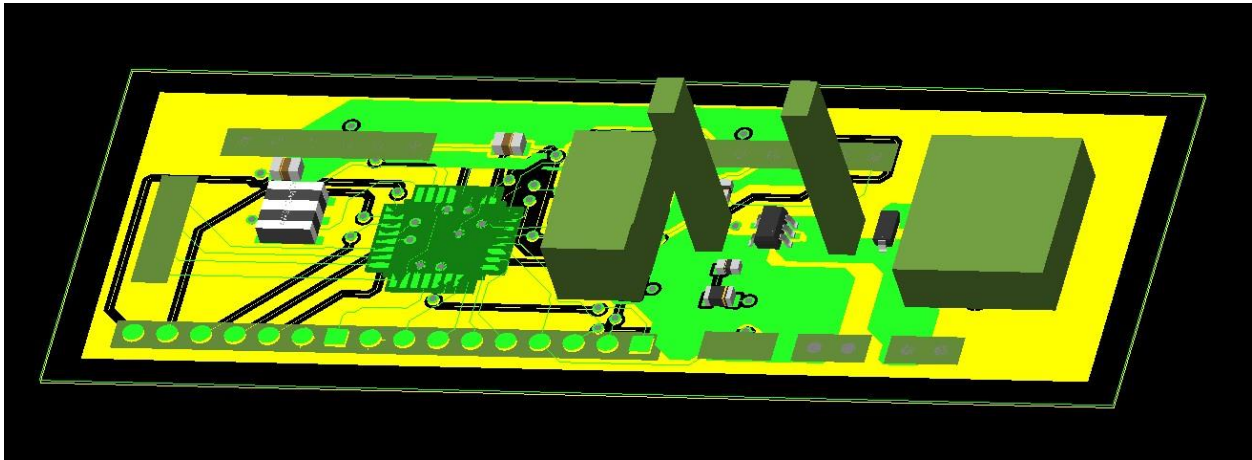


Bottom layer complementary soldermask

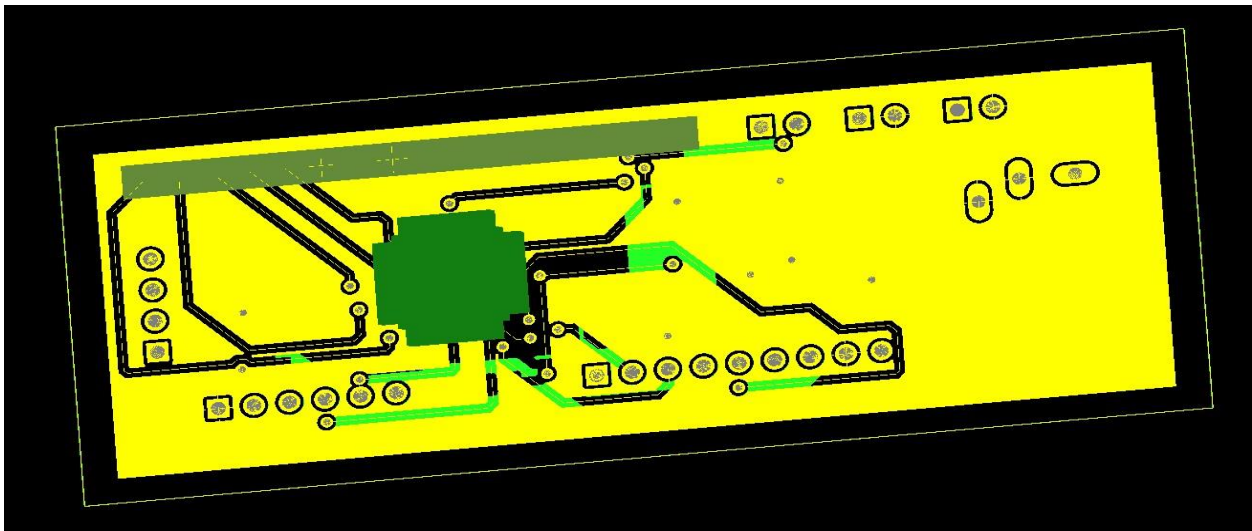


Pastemask top layer

Infine ho generato una view 3D della scheda realizzata:



Top 3D view



Bottom 3D view

Nella top view effettivamente sembrano mancare alcuni componenti che non sono chiaramente identificabili, probabilmente per un errata configurazione del tool di 3D view in quanto quest'ultimo step non era stato effettuato nell'esperienza di laboratorio e ho provveduto autonomamente per la sua generazione, ma tutto sommato ci dà un'idea di come sarà la nostra PCB una volta realizzata.

5 Conclusioni

Il progetto ci ha appreso di imparare i rudimenti per la progettazione di PCB dual layer; le conoscenze apprese sono tranquillamente valide e trasferibili in qualsiasi altro software, tipo Altium o KiCad.